Representing Temporal Patterns in Computer-Interpretable Clinical Guidelines

António Silva[1], Tiago Oliveira[2], Paulo Novais[2], and José Neves[2]

1 **Department of Informatics, University of Minho**
Braga, Portugal
pg25307@alunos.uminho.pt

2 **Algoritmi Research Centre/Department of Informatics, University of Minho**
Braga, Portugal
{toliveira,pjon,jneves}@di.uminho.pt

Abstract

Computer-Interpretable Guidelines (CIGs) as machine-readable versions of clinical protocols have to provide appropriate constructs for the representation of different aspects of medical knowledge, namely administrative information, workflows of procedures, clinical constraints and temporal constraints. This work focuses on the latter, by aiming to develop a comprehensive representation of temporal constraints for machine readable formats of clinical protocols and provide a proper execution engine that deals with different time patterns and constraints placed on them. A model for the representation of time is presented for the CompGuide ontology in Ontology Web language (OWL) along with a comparison with the available formalisms in this field.

1998 ACM Subject Classification I.2.4 Knowledge Representation Formalisms and Methods, J.3 Life and Medical Sciences - Medical Information Systems.

Keywords and phrases Computer-Interpretable Guidelines, Temporal Constraints, Clinical Decision Support, Ontologies

Digital Object Identifier 10.4230/OASIcs.xxx.yyy.p

1 Introduction

CDSSs providing patient-specific recommendations follow a more consulting style of communication and require substantial modelling activity [8]. The models used to represent medical knowledge in such systems range from probabilistic models and decision trees to task-network models (TNMs). The latter are arguably the most used, mainly because they allow the representation of chains of events and a wide variety of situations [7, 11]. TNMs are the basis for Computer-Interpretable Guidelines (CIGs), machine readable versions of clinical protocols. There are many CIG models available, but they have yet to overcome, for the most part, the stage of academic project. As a result, there is no standard for the representation of CIGs. There are, however, influential CIG approaches such as GLIF3 [3], PROforma [6], Asbru [12], SAGE [15], and GLARE [14]. The downside of this is each model tends to focus solely on one aspect of the representation of clinical protocols, disregarding, or not paying as much attention to, the other important aspects. In the representation of CIGs, according to [16], one should take into account: the representation of administrative information, the construction of workflows of clinical procedures, the representation of clinical constraints and patient state conditions, and the representation of temporal constraints. The CompGuide project [9] aims to build a comprehensive ontology for CIGs which gathers the main strengths of the existing approaches. It explores Ontology Web Language (OWL) as the

Conference/workshop/symposium title on which this volume is based on.
Editors: Billy Editor and Bill Editors; pp. 1–8

OpenAccess Series in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

support language for the definition of representation primitives and the procedural logic of clinical protocols. The work presented in this paper concerns the representation of temporal patterns within CompGuide, in a way that balances complexity and expressiveness. The representation of temporal patterns conveyed in clinical protocols is quite complex, firstly because they may appear in various textual expressions, such as in the expression "history and physical every 3-6 months for 2 years, then every 6 months for a total of 5 years" extracted from a clinical protocol for the treatment of colon cancer [2], and secondly because they be extremely intricate. As such, temporal model is proposed for the definition of temporal constraints and the management of time in medical algorithms. At the same time, it is a complementary model to the existing approaches, by gathering the most developed aspects of each approach and filling in the gaps that often render said approaches incomplete. The main contribution of this model is an integrated representation of durations, periodicities, stop conditions for clinical tasks and temporal restrictions for conditions about the state of a patient, which is something the existing CIG models have not achieved.

The paper is organized as follows. Section 2 provides related work about the representation of temporal constraints in CIG models. Some background on the CompGuide ontology is presented in section 3. The temporal model along with examples for discussion are provided in section 4. Finally, section 5 presents conclusions about the work developed so far and future work considerations.

2 Related Work on the Temporal Representation of Clinical Protocols

The management of time is one of the main concerns in CIG modelling, as clinical processes are chains of events unfolding over time. A few CIG approaches have been specifically devised to deal with temporal constraints. GLIF3 [3] deals with both temporal constraints placed on patient state conditions and durations of actions. Asbru [12] provides a comprehensive representation model for durations as well. In fact, this CIG model presents clinical protocols as time-oriented skeletal plans for which it is possible to define time annotations, which may be constraints on the starting time and ending time of tasks (such as earliest possible start and earliest possible ending), maximal and minimal durations, and cyclical time points (e.g., every morning, every day, etc.). A step further is given in GLARE [13], which introduced the representation of periodicities for events and repetition schemes. This formalism was later expanded in [1]. The new version of the work provides an enhanced formalism to express periodicities, with the possibility of defining delays between the cycles of the periodic event. It also became possible to define more complex periodicity periodicity patterns. For instance, each cycle of a periodic event may have itself an associated periodicity. Another interesting development is the mapping of the high-level time patterns to a Simple Temporal Problem (STP) [4] reasoning framework, in which a graph representation of the guideline, resulting from the calculation of the relative time constraints of tasks, is provided. Despite being one the most used temporal reasoning AI techniques, the authors mention they were unable to represent complex time patterns with it, as the STP-tree structure they produce is not suited for events that repeat over time. A similar approach is adopted in [5], but in this case the temporal representation was tailored for clinical plans in the oncology. This reflects the dominant view of pure AI approaches [10], stemming mainly from logic, being faced with obstacles when applied to the medical domain. In both Asbru and GLARE it is not possible to express temporal constraints about clinical parameters of the state of the patient.

Temporal constraints in clinical protocols can be divided into two groups. One consists of the constraints placed on conditions about the state of a patient in order to express for

how long a clinical parameter holds a certain value. The other group consists of constraints placed on the execution of tasks. In this group, one can have qualitative and quantitative constraints. The former are expressed through the control relationships in the guideline and represent the relative order of tasks. The latter include patterns such as durations, delays, periodicities and repetitions. As far as our knowledge goes there is a lack of an approach encompassing all these aspects of temporal constraints.

3 The CompGuide Ontology

The CompGuide [9] ontology provides a task network model representation for clinical protocols in OWL. In order to fulfill this purpose, it follows a logic in which complex information elements are represented as instances with multiple object properties connecting them to other instances, and simple information which cannot be further decomposed is represented using data properties. However, simple information that is reusable and will most likely be needed across different guidelines is represented as instances from specific classes as well. In this regard the representation is similar to a linked list of procedures.

As such, a clinical protocol is represented as an instance of the *ClinicalPracticeGuideline* class. Individuals from this class have a set of data and object properties allowing the representation of descriptive and administrative guideline information such as the name of the guideline, its general description, date of creation and last update, version, clinical specialty, category, intended users, and target population. An example of the initial definition of a guideline is given in Fig. 1. The guideline is the NCCN Clinical Practice Guideline for Colon Cancer [2], one of the case studies we are developing.

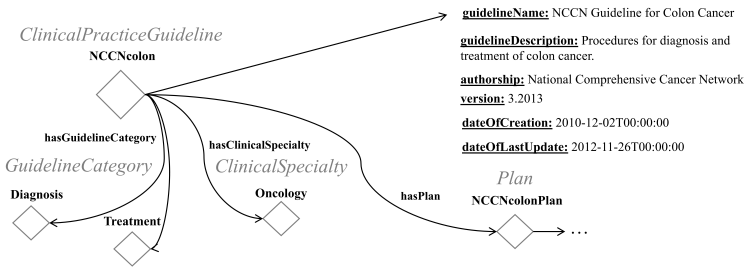

Figure 1 Initial definition of a National Comprehensive Cancer Network guideline for the treatment of colon cancer in the CompGuide ontology.

Every instance representing a guideline is linked to an instance from the class *Plan*, which is a container of tasks, a complex task. In turn, an instance from *Plan* is linked to other instances symbolizing basic tasks. These basic tasks are represented using three classes: *Action*, *Decision* and *Question*. The objective here is to create a recommendation plan containing references to specific types of tasks. The *Action* class expresses a procedure should be carried out by a health care professional. There are several subtypes of actions in the ontology specifying their nature with more detail. The *Decision* class is used to make assertions about the state of the patient, to infer new information from the existing one. The most obvious example of such a task is clinical diagnosis. The *Question* task is used to get information about the symptoms, health condition or other parameters that may help to characterize the state of a patient. *Questions* are also used to register information from the observations of the physician, and to store results from clinical exams. This type of task

gathers all the information necessary for the execution of a clinical protocol. Through object properties, it is possible to define the different control relations that may exist between tasks, the sequence of execution of tasks or if they should be executed simultaneously or concurrently. Regarding this, it is possible to define the sequential execution of tasks, the parallel execution of tasks, points in which one of various alternative tasks is selected (either automatically or by choice of the user), and careflow synchronization points. Additionally, the ontology provides different types of clinical constraints. From simple conditions determining the selection of an option in a decision task, to trigger conditions, which are used to selected a task from a list of alternatives. It is also possible to express pre-conditions, which are requirements to be verified before the execution of a task, expected outcomes of tasks in terms of changes in the state of a patient, and repetition conditions for clinical tasks. Comparatively, CompGuide does not require proficiency in a constraint programming language in order to define these conditions, unlike the existing approaches. Moreover, it provides an increased expressiveness in the definition of tasks and control relationships.

4 Proposal for a Temporal Representation Model

The definition of temporal representation primitives follows the logic described in section 3. The classes of the temporal model are shown in Figure 2. The main classes are represented as subclasses of *TemporalElement*. One of those subclasses is *TemporalUnit*, which represents the different granularities a temporal constraint may have, such as *second*, *minute*, *hour*, *day*, *week*, *month*, and *year*. The control relations mentioned in section 3 are responsible not only for the establishment of a workflow of tasks, but also for the definition of qualitative temporal constraints, i.e., the relative order of tasks within the guideline. At runtime, a guideline execution engine analyses these constraints and builds a map of task execution. As this is an aspect already included in the ontology and in conformity with existing approaches, this section is concerned mainly with quantitative temporal constraints.

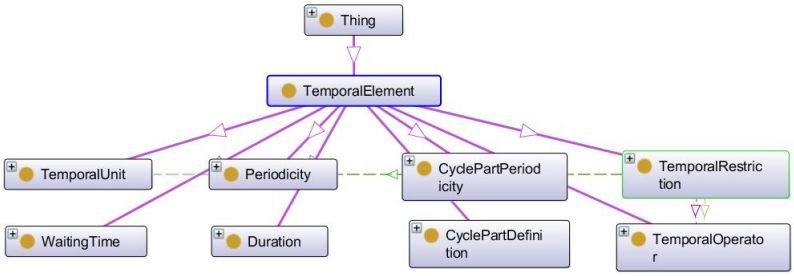

Figure 2 Classes of the model for the representation of temporal constraints in the CompGuide ontology.

4.1 Temporal Constraints on the Execution of Tasks

Expressing for how long a task should be executed is one of the main temporal patterns in clinical protocols. In the proposed model this possible with the *Duration* class. The attributes characterizing this class are encoded as necessary conditions in OWL (as it happens with all the other classes). As such, to define a duration, one should choose either to define a maximal and minimal duration, trough the *maxDurationValue* and *minDurationValue* data properties, or to define an exact value for the duration, through the *durationValue* data

property. The range of these data properties are decimal numerical values. Regardless of the type of value one defines, it is always necessary to define a temporal granularity for a decision, which is done through the *hasTemporalUnit* object property which links instances of *Duration* to instances of *TemporalUnit*. Compared with Asbru [12], this form of temporal representation is simpler because it does not feature annotations about the earliest and latest start and ending times of tasks. Such a simplification may be regarded as limitation, yet the objective was to meet the temporal restrictions conveyed in clinical protocols for which the temporal constructors of Asbru may be excessive and impractical. Durations are defined for *Actions* and *Plans* since these are the only tasks which can be executed over a certain time. In fact, in most cases, the information about the duration of tasks is conveyed either as an exact value or as an interval, such as in the case of the natural language expression "perform neoadjuvant therapy for 2-3 months" extracted from [2]. This removes the need for complex time annotations. However, it is still possible to express delays between tasks with the class *WaitingTime*, whose values (max, min, exact) and temporal granularity are defined in a similar way as in *Duration*. Delays can be defined for all classes of tasks in CompGuide, allowing the representation of situations such as the *Action* "re-evaluation for cancer resection after 2 months of preoperative chemotherapy", in which there is a "re-evaluation" task which should be delayed for "2 months", after the end of "preoperative chemotherapy". However, the proposed model provides a better definition of intervals and exact values. For instance, in [1], durations are always represented as intervals. To express exact durations, it is necessary to state the upper and lower bounds of the interval are the same. By having a data structure for exact durations, one can make the processing of such constraints simpler.

The representation of periodic tasks is the most complex pattern. In CompGuide, this pattern is represented by the class *Periodicity*. A periodicity can be defined for any type of task. However, the periodic event is bound by either a duration, a repetition constraint or a stop condition about the state of the patient. The duration is defined through the reuse of the *Duration* class. As such, an instance of *Periodicity* can also be linked to an instance of *Duration* through the *hasDuration* object property, thus determining for how long a periodic event should take place. On the other hand, if one wants to state the number of times the event should be carried out (the same is to say the number of cycles of the periodic event), it is necessary to formulate a repetition constraint, which is possible through the *repetitionValue* data property with a range of integer numerical values. Alternatively, it could be the case the periodic task should only occur until a condition about the state of a patient is met. To express this, one uses the *hasStopCondition* object property to link an instance of periodicity to instances of the class *StopCondition*. While it is possible for a periodicity to have a duration and a stop condition, a repetition value and a stop condition, or just a stop condition, it is not possible to have both a duration and a repetition value for it is considered to be redundant information. With a duration and a frequency it is already possible to calculate the number of repetitions of an event and vice versa. The stop condition takes precedence over the other temporal restrictions, so if the condition is met, the task is immediately stopped.The frequency of the periodicity and the temporal granularity are defined in the data property *periodicityValue* and through the *hasTemporalUnit* object property, respectively. A cycle of a periodic event may have itself an associated periodicity or duration. In order to represent this, the object property *hasCyclePartDefinition* is used. It links instances of *Periodicity* to instances of *CyclePartDefinition*, where one actually defines the periodicity of the cycle or its duration. In the class *CyclePartDefinition*, the periodicity of the cycle is defined through the object property *hasCyclePartPeriodicity*, which links instances of *CyclePartDefinition* to instances of *CyclePartPeriodicity*. *CyclePartPeriodicity* is different

from the main *Periodicity* class. This intricate representation of periodic tasks is depicted in Figure 3 in order to convey the distinction between *Periodicicty* and *CyclePartPeriodicity*. One can argue it would be simpler to reuse the *Periodicity* class rather than defining another class for the definition of the periodicity of each cycle, but by doing so it would be possible to nest periodicities inside one another infinitely, which does not quite fit the cases appearing in clinical protocols. Cancer treatment guidelines are usually very rich in periodicity temporal patterns, mainly because of the chemotherapy/radiotherapy regimens they recommend. An action representing this is the following for CapeOX chemotherapy: "CapeOX every 3 weeks with the administration of capecitabine twice daily for 14 days" [2]. In the statement, it is possible to distinguish the *Periodicity* (every 3 weeks) from the *CyclePartPeriodicity* (twice daily for 14 days). In terms of expressiveness, this approach allows the representation of temporal constraints which are not representable in Asbru, which is one of the dominant CIG models. It is, at the same time, an adaptation (mainly in the periodicity components) of the formalism presented in [1].

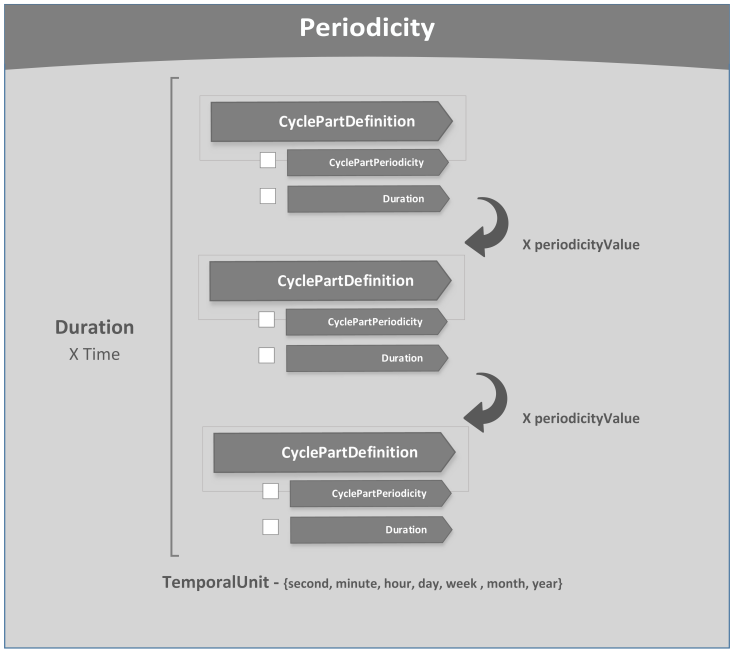

Figure 3 Schematic representation of different levels of periodicity for a task, with a distinction between *Periodicity* and *CyclePartPeriodicity*.

4.2 Temporal Constraints on the State of a Patient

Most CIG approaches represent temporal constraints for the conditions about the state of a patient as strings in description fields. Considering the different conditions in CompGuide, such as conditions for decisions, trigger conditions, expected outcomes, repetition types of conditions, and pre-conditions, it would be advantageous to develop a form of automated reasoning about them, thus making necessary the development of a structured way to represent them.

In CompGuide, a temporal constraint for conditions about the state of a patient is represented by an instance of the *TemporalRestriction* class. To link the constraint to an

instance representing a condition, it is necessary to use the *hasTemporalRestriction* property, which, although non-mandatory, can be asserted for any of the above-mentioned conditions. For each instance of *TemporalRestriction* it is necessary to specify a temporal operator through the *hasTemporalOperator* object property. This object property points to individuals of the class *TemporalOperator*. This is an enumerated class that can only have a limited number of instances, namely the following: *currently*, *within_the_last*, *during*, *within_the_following*. The temporal operators represent the reach of a temporal constraint and are coupled with temporal ganularities, defined through the *hasTemporalUnit* object property, and temporal restriction values. The latter are expressed through data properties such as *maxTemporalRestrictionValue* and *minTemporalRestrictionValue*, or *temporalRestrictionValue* for exact values. The operator *currently* expresses the condition must hold at the time of execution, when the clinical task is being considered for implementation. On the other hand, *within_the_last* is used when one wants to express a condition must have held true at least once, within a period of time just before execution time. But, if the intention is to state that a certain condition must have to continuously hold true during a period of time just before the execution time, then one should use the operator *during*. These three temporal operators fit in temporal restrictions of conditions for decisions, trigger conditions, repetition conditions, and pre-conditions because these are used to reason about the present or the past. Yet, in an expected outcome, it is necessary to express a condition about the future, in which one expects to observe the effect a clinical task has after being applied to a patient. For such, it is possible to use the operator *wihin_the_following*, which bounds the observation of the condition to a certain period in the future, starting from the time of execution. An example of a temporal restriction using a temporal operator is an expected outcome of a chemotherapy regimen, such the statement "the tumor should become operable within 6-7 months of FOLFOX or CapeOX chemotherapy", extracted from [2].

5 Conclusions and Future Work

Besides the need to further evaluate the expressiveness of the model with an array of case-studies containing a wide variety of temporal patterns, it is necessary to focus on another important aspect of CIG temporal representation, which is interpretation. CIG interpretation can happen in two distinct moments: acquisition and execution. During acquisition, the objective of the interpretation is to check the consistency, which in this case is the temporal consistency, of a guideline, and during execution, the objective is to provide timely recommendations. In both situations, the treatment of temporal constraints goes beyond the isolated processing of each constraint. Instead, it is necessary to take into account the so called part-of relations, which exist when an atomic task is inside a complex task such as *Plan*. In such case, and just to give an example, the combined durations of the tasks inside a *Plan* cannot be greater than the duration of the *Plan* itself. A time manager is currently under development for the CompGuide ontology. The next steps include the evaluation of the tractability, correctness and completeness of this approach. The presentation of the temporal plans during execution will have the form of workflows with timely notifications about when the tasks should be executed and how long they should last.

Acknowledgements This work is part-funded by ERDF - European Regional Development Fund through the COMPETE Programme (operational programme for competitiveness) and by National Funds through the FCT - Fundação para a Ciência e a Tecnologia (Portuguese Foundation for Science and Technology) within project FCOMP-01-0124-FEDER-028980

and project Scope UID/CEC/00319/2013.The work of Tiago Oliveira is supported by a FCT grant with the reference SFRH/BD/85291/ 2012.

References

1 Luca Anselma, Paolo Terenziani, Stefania Montani, and Alessio Bottrighi. Towards a Comprehensive Treatment of Repetitions, Periodicity and Temporal Constraints in Clinical Guidelines. *Artificial Intelligence in Medicine*, 38(2):171–195, 2006.

2 Al Benson, Tanios Bekaii-Saab, Emily Chan, Yi-Jen Chen, Michael Choti, Harry Cooper, and Paul Engstrom. NCCN Clinical Practice Guideline in Oncology Colon Cancer. Technical report, National Comprehensive Cancer Network, 2013.

3 Aziz a Boxwala, Mor Peleg, Samson Tu, Omolola Ogunyemi, Qing T Zeng, Dongwen Wang, Vimla L Patel, Robert a Greenes, and Edward H Shortliffe. GLIF3: A Representation Format for Sharable Computer-Interpretable Clinical Practice Guidelines. *Journal of Biomedical Informatics*, 37(3):147–61, June 2004.

4 Rina Dechter, Itay Meiri, and Judea Pearl. Temporal constraint networks. *Artificial intelligence*, 49(1):61–95, 1991.

5 Juan Fernandez-Olivares, Luis Castillo, Juan a. Cózar, and Oscar García Pérez. Supporting clinical processes and decisions by hierarchical planning and scheduling. *Computational Intelligence*, 27(1):103–122, 2011.

6 John Fox and Richard Thomson Ma. Decision Support for Health Care : The PROforma Evidence Base. *Informatics in Primary Care*, 14(1):49–54, 2006.

7 D Isern and A Moreno. Computer-based Execution of Clinical Guidelines: a Review. *International Journal of Medical Informatics*, 77(12):787–808, 2008.

8 Mark A. Musen, Yuval Shahar, and Edward H. Shortliffe. Clinical decision-support systems. In EdwardH. Shortliffe and JamesJ. Cimino, editors, *Biomedical Informatics*, Health Informatics, pages 698–736. Springer New York, 2006.

9 Tiago Oliveira, Paulo Novais, and José Neves. Representation of Clinical Practice Guideline Components in OWL. In *Trends in Practical Applications of Agents and Multiagent Systems SE - 10*, volume 221 of *Advances in Intelligent Systems and Computing*, pages 77–85. Springer International Publishing, 2013.

10 A. K. Pani and G. P. Bhattacharjee. Temporal representation and reasoning in artificial intelligence: A review. *Mathematical and Computer Modelling*, 34(1-2):55–80, 2001.

11 Mor Peleg. Computer-interpretable Clinical Guidelines: A Methodological Review. *Journal of Biomedical Informatics*, 46(4):744–63, 2013.

12 Y Shahar, S Miksch, and P Johnson. The Asgaard Project: A Task-specific Framework for the Application and Critiquing of Time-oriented Clinical Guidelines. *Artificial intelligence in Medicine*, 14(1-2):29–51, 1998.

13 Paolo Terenziani, Carlo Carlini, and Stefania Montani. Towards a comprehensive treatment of temporal constraints in clinical guidelines. In *Proceedings of Ninth International Symposium on Temporal Representation and Reasoning 2002*, pages 20–27. IEEE, 2002.

14 Paolo Terenziani, Stefania Montani, Alessio Bottrighi, Mauro Torchio, Gianpaolo Molino, and Gianluca Correndo. The GLARE Approach to Clinical Guidelines: Main Features. *Studies in Health Technology and Informatics*, 101(3):162–6, 2004.

15 S W Tu, J R Campbell, J Glasgow, M A Nyman, R McClure, J McClay, C Parker, K M Hrabak, D Berg, T Weida, and Others. The SAGE Guideline Model: Achievements and Overview. *Journal of the American Medical Informatics Association*, 14(5):589, 2007.

16 Dongwen Wang, Mor Peleg, Samson W Tu, Aziz A Boxwala, Robert A Greenes, Vimla L Patel, and Edward H Shortliffe. Representation primitives, process models and patient data in computer-interpretable clinical practice guidelines: a literature review of guideline representation models. *International Journal of Medical Informatics*, 68(1-3):59–70, 2002.